

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.**

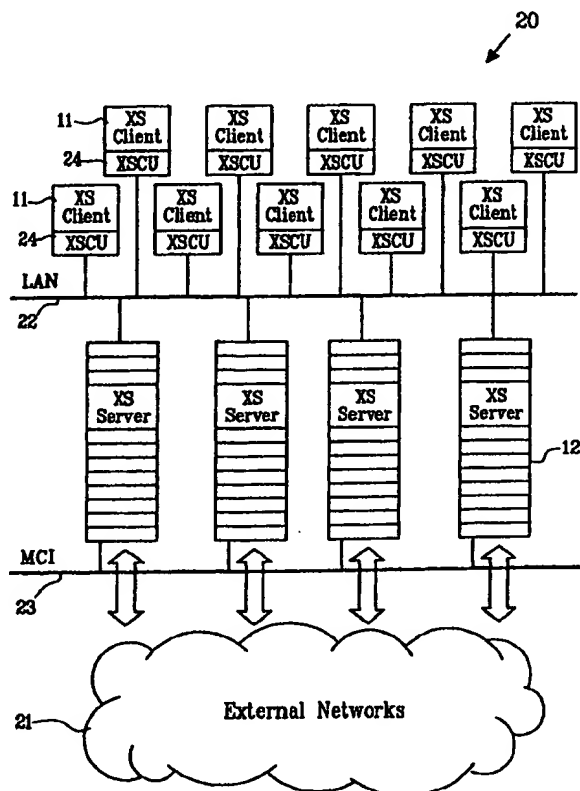


## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<b>(51) International Patent Classification <sup>7</sup> :</b> <b>G06F 17/30, H04L 12/00 // G06F 9/46</b>	<b>A1</b>	<b>(11) International Publication Number:</b> <b>WO 00/33214</b> <b>(43) International Publication Date:</b> 8 June 2000 (08.06.00)
<b>(21) International Application Number:</b> PCT/SE99/02247 <b>(22) International Filing Date:</b> 1 December 1999 (01.12.99) <b>(30) Priority Data:</b> 9804188-2                      1 December 1998 (01.12.98)                      SE <b>(71) Applicant:</b> TELEFONAKTIEBOLAGET LM ERICSSON [SE/SE]; S-126 25 Stockholm (SE). <b>(72) Inventor:</b> BORNSTEIN, Jan; Rosenhillsgatan 2, S-412 63 Göteborg (SE). <b>(74) Agent:</b> GÖTEBORGS PATENTBYRÅ DAHLS AB; Sjöporten 4, S-417 64 Göteborg (SE).		<b>(81) Designated States:</b> AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).  <b>Published</b> <i>With international search report.          Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>

**(54) Title:** METHOD AND ARRANGEMENT RELATING TO RESOURCE HANDLING IN COMMUNICATIONS NETWORKS**(57) Abstract**

The present invention relates to a method and an arrangement in a communications network including one or more clients (11) and one or more servers (12) providing services and/or resources (13). The client (11) accesses at least one server for obtaining said resource or service (13). Said arrangement includes a part (XSCU) (24) arranged for controlling communication between the client and the server (12), said part (24) including an updatable database (XSCU Context) for storing information at least about said resource/service and corresponding server (12).



**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

Title

**METHOD AND ARRANGEMENT RELATING TO RESOURCE HANDLING IN  
COMMUNICATIONS NETWORKS**

5

**TECHNICAL FIELD OF THE INVENTION**

The present invention relates to a method and an arrangement in a communications network,  
specially in a computer network including one or more client stations and one or more servers,  
10 arranged to provide services and/or resources, when the client station accesses at least one  
server for obtaining a service or resource.

**BACKGROUND OF THE INVENTION**

15 In a network including communication apparatuses, such as computer networks or  
telecommunication networks, usually one or several clients (e.g. workstations, subscribers etc.)  
are connected to one or several service provider stations (servers, base stations etc.). The client  
sends messages to obtain resources. The resource can be a specific resource, preferably  
identified by a Globally Unique IDentifier (GUID) without a reference to any specific server.

20

A resource may be any kind of software or hardware resource, such as procedures or functions  
controlling specific routines, switching functions etc. belonging to the first category, and print  
services, network facsimile services, CD-ROM racks/Jukeboxes, communication  
ports/boards/paths etc. belonging to the second category.

25

Presently, in some systems a Resource Manager (RM), which is a global resource, is arranged  
to keep track of all resources included in the system. Moreover, the RM determines which  
server contains a specific resource. Such systems incorporate drawbacks: the RM will become a  
bottle neck in the system when the number of the resources and servers increases; and the  
30 system will slow down when simultaneous multiple accesses are made to determine resources  
and corresponding servers.

Japanese Patent Application No. 195972, describes a single client/single server system including a resource management server, which has a first internal memory. Said memory stores the name of a server which outputs a Resource Output Request and the resource name. There is also included a resource updating unit. The updating unit searches the contents of the first internal memory during an update period and an updating notice is sent to the resource server. Access to the free resources is then managed by the server.

In another solution when a client requests a resource, a client resource request message is sent to an arbitrary server. If said server does not own or has access to the resource, the message is sent to next the server. The message transmission is continued until a server owning the resource is found. Apparently, this solution will generate a large amount of messages, which eventually will slow down the system.

Japanese Patent Application No. 164837, describes a distributed type information processing method, e.g. for a computer system involving executing job demanded by a client through a server based on information related to hardware and software resources. The method involves referring a program module when the execution of a specific job is demanded by a client. Information related to hardware resources is collected by an agent. Moreover, information related to software resources is also collected. Then a server is arranged to execute the demanded job based on the collected information.

US 5,548,723 relates to object-oriented client-server facility (CSF) and networking service facility (NSF) interfaces, which implement communication between application programs residing in client and server nodes of a distributed services network. The CSF interface includes remote procedure call (RPC) objects for invoking and responding to service requests at the nodes, and application programming interface (API) objects for transporting those requests between the nodes. However, the API objects only provide communication transports within a node. Accordingly, the API and RPC objects interact with dynamically-configurable protocol stacks within the NSF interfaces to complete the transport mechanism needed by an application program on the client node when accessing services on a remote server node.

US 5,101,348 concerns a communications network, each network node can maintain its own list of network resources in a topology database. When the state of a resource "owned" by a particular node changes, that node broadcasts a topology database update (TDU) message to adjacent nodes. Each adjacent node updates its own topology database and rebroadcasts the message. To minimize the amount of information that must be included in TDU messages when two nodes are reconnected after an outage, each node assigns flow reduction sequence numbers (FRSNs) to TDU messages and keeps a record of the FRSN for the last TDU message sent to an adjacent node. The node also records, for each resource in its database, the FRSN of the last TDU message including that resource. When two nodes are reconnected, the sending node includes in the TDU message only those resources having a FRSN greater than the FRSN assigned to the last TDU sent to the adjacent node to which the TDU message is directed.

Non of above documents relates to an arrangement includes a client API part arranged for controlling communications between the client and a server, which part includes an updateable database for storing information about a resource/service and to said resource/service corresponding server . Moreover, according to US 5,548,723, the API objects (not client API) only provide communication transports within a node.

## SUMMARY OF THE INVENTION

20

The main object of the present invention is to provide an arrangement that when a client in a network sends a request for some specific action the arrangement will through a user interface will rout the client messages to an appropriate server in the network.

25 One object of the present invention is to provide a rigid communication system which manages the resource distribution between included devices in a simple and less complex way known through prior art.

Another object of the present invention is to provide an arrangement to overcome above-mentioned problems and drawbacks.

30

Yet another object of the present invention is to provide a network arrangement that will distribute resources even if parts of the network are out of order.

5 Still another object of the present invention is to provide a network arrangement that will provide resources for a client without the client having any specific knowledge of the resource providing elements (servers).

Therefore the arrangement according to the beginning includes a part arranged for controlling communication between the client and the server, said unit including an updatable database for  
10 storing information at least about said resource/service and corresponding server. Preferably, said part is a client API or implemented as a distributed, asynchronous, telegram-based client API. In one embodiment said part is arranged to distribute client resource/service requests to said servers by managing the database and to achieve a safe system.

15 In one advantageous embodiment said part also includes server configuration, allowing multi-node switching, load-sharing and error handling, and a robust network is obtained since said error handling is based on the "graceful degradation" paradigm, where an error just affects the functionality close to the failure.

20 Preferably, said resource or service is a specified and identified with a Globally Unique Identifier and the database contains information about all resources used by all clients and also their location.

The invention also refers to a communications network, which further comprises networking  
25 arrangement connecting client stations and server units. The client stations are arranged to generate messages for requesting services and/or resources from the server units. At least each server unit handles a resource/service. The interfacing means is a service access-point among others including server unit configurations and resource/service states. Advantageously, said interfacing means allows multi-node switching, load-sharing and error handling and it is  
30 implemented as a distributed, asynchronous, telegram-based client API. The server unit is also arranged to handle interface means as a resource/service.

To obtain a stable network, the network architecture is substantially a decentralized distributed system and based on a varying number of server units connected to each other via a Local Area Network (LAN), for example via an IP network such as an ethernet LAN. In the network the server units are also connected to each other via a network to achieve switching  
5 between the server units. Preferably, the network is configured using one server unit with a varying number of clients. It may also be configured as a distributed multi-server system providing a varying number of clients, to be connected to a varying number of server units.

In the network Hardware (HW) interfaces are handled as resources, which are encapsulated  
10 under a generic resource interface Manager within a server unit and different resource modules are encapsulated and handled as subsystems under the resource interface.

In an advantageous embodiment the resource modules include at least one of ISDN Resource module, for handling ISDN (Integrated Services Digital Network) protocol, S-Connect  
15 Resource module, for handling an analogue operator interface (S-connect), DSP Resource module, for handling DSP (Digital Signal Processing) handling for signalling and messaging, and Conference Resource module, for handling conference devices for conference switching.

Moreover, the functions of the server unit are managed by a Request Manager Unit (XSRMU),  
20 Switch Manager unit (XSSM) Resource Manager Unit (XSRM), Main Manager (XSMM) and resource connection. XSRM provides functionality to manage, synchronise and co-ordinate HW resources, XSSM handles the resources in a logical level and manages the relations between the resources and uses the XSRM to access them, XSMM contains main application and initiates and synchronises other units in the system, and XSRMU provides functionality to handle the  
25 communication in the distributed environment.

In a most preferred embodiment the network is an Exchange System based on computers, preferably Personal Computers (PCs).

30 Consequently, the method according to the present invention comprises the steps of: providing means for controlling communication between the client and the server, said means including



an updateable table (XSCU Context) for storing information at least about said resource/service and corresponding server, controlling said table when sending a request for said resource/service, and based on information retrieved from said table, routing said request to a server suited to handle the request. In the server, the client requests are received by a Request Manager, which handles the server communication part, and depending on the type of the request, the Request Manager distributes the requests to a Switching manager or Resource Manager via a main queue. Preferably, the switching is multi-node and provided in several stages, including: (a) allocation of a connection means resource and sending a Message (XSMMSG) to other servers requesting an execution of a service and all information needed by the other server to fulfill its part of the switching, (b) when another server receives the Message (XSMMSG), connecting the input from the connection means to an output on a relevant connection point, (c) sending an XSMMSG containing a service execution message and all information needed by the originating server to fulfill stage (b), and (d) when the originating server receives a Service Execution Message, connecting the input connection point to an output on the connection means.

Moreover, to minimize the network load according the method the client request creates new resources, which are arranged in an arbitrary server queue to distribute the workload over the servers and if this request contains a pointer, transferring the request to a server pointed out by the pointer.

#### BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will become more fully apparent from the claims and the description as it proceeds in connection with the drawings, in which:

- Fig. 1 is a very schematic block diagram showing the main elements of a network involving an arrangement according to the invention;
- Fig. 2 illustrates schematically a network embodiment including the invention; and
- Fig. 3 is a more detailed but still schematic illustration of the elements of the network arrangement according to fig. 2.

## DETAILED DESCRIPTION OF AN EMBODIMENT

According to a preferred embodiment, shown in fig. 1, in a communications network 10, preferably a multi client 11/multi server 12 environment, a client (station) 11 requiring a specific resource 13 sends a message referring to said specific resource, which is identified with a Globally Unique IDentifier (GUID). Preferably, the request does not refer to any specific servers 12 in the network. Consequently, the client does not need to have any special knowledge about the servers in the network. Preferably, the messages are queued in message queue systems 14. The queue systems 14 will be described later.

According to the invention, an information database or a context table is created, which is linked to a client API (Application Programmers Interface). The database or resource context is arranged to include information about and keep track of substantially all resources in the network. The resources may be of the type that can be created and/or destroyed. API, by definition is, a set of routines that an application uses to request and carry out lower-level services performed by a computer's operating system.

In the following, the invention will be described in more detail in connection with a non-limiting embodiment given merely to facilitate the better understanding of the invention.

The embodiment, schematically shown in fig. 2, relates to an Exchange System (XS) 20, which is a computer based communications platform providing distributed and redundant functionality for switching, control and coordination of different communication protocols interfacing external (or internal) networks 21. The external et works can for example be any of PLMN (Private Land Mobile Network), PSDN (Public Switched Data Network), PSTN(Public Switched Telephone Network) etc.

The XS architecture is substantially a decentralized distributed system and can be based on a varying number of XS servers 12 connected to each other, for example via an IP network such as an ethernet LAN (Local Area Network) 22.

The XS servers 12 may also be connected to each other via a network 23, for example through MC1 bus, to achieve switching between the servers.

The XS 20 can be configured using one XS server with a varying number of clients 11, or as a distributed multi-server system, that provides a varying number of clients, to be connected to a varying number of XS servers.

In the system, the Hardware (HW) interfaces are handled as resources, and can be encapsulated under a generic resource interface XS Resource Manager within a XS server 12. Preferably, different, e.g. protocol specific, resource modules are encapsulated and handled as subsystems under the resource interface. The XS is therefore extendable to handle new types of resources, e.g. communication protocols and networks in future versions, e.g. IP-telephony and different radio system protocols.

15 Mainly, the XS has the following functions:

- Switching, which covers the generic, protocol independent switching.
- Call Control, which covers protocol independent call control functionality.
- Signalling, which covers signal independent tone sending/receiving.
- Messaging, which covers message independent message handling, such as playing/recording.

To obtain these functions, there are some additional functional areas:

- Server connection, which handles the connection between the CL (clients) and XS.
- Resource handling, which handles resource allocation, e.g. conference allocation.
- 25 - Status notification, which handles asynchronous XS and external network status information to the clients.
- System management, which handles XS configuration, monitoring, control etc.

The XS is based on implementation of a number of resource modules, such as:

- 30 - ISDN Resource module, which handles the ISDN (Integrated Services Digital Network) protocol.

- S-Connect Resource module, which handles the analogue operator interface (S-connect).
- DSP Resource module, which handles the DSP (Digital Signal Processing) handling for signalling and messaging.
- Conference Resource module, which handles the conference devices for conference switching.

A DSP is a resource that can be allocated/deallocated and used for tone and message sending/receiving. A conference resource provides functionality to set up a set of participants represented as seats into a forum, where the participants are connected. The connection to a conference seat can be a simplex or a duplex connection represented.

The XS Communication Unit (XSCU) API 24 is the name of the XS service access-point. Preferably, it is implemented as a distributed, asynchronous, telegram-based client API.

- 15 The XSCU API includes among others the XS server configuration and allows multi-node switching, load-sharing and error handling, for example based on the "graceful degradation" paradigm, i.e. degradation of the system in such a manner that it continues to operate, but provides a reduced level of service rather than failing completely and an error just affects the functionality close to the failure.

20

The Client Logic (CL) Interface to the XS is the XSCU API, which is included in the client application. The XSCU handles the communication between the client and the XS and includes functionality to distribute client requests to the XS servers, by managing client context information.

25

- Fig. 3 shows the components involved in the system. The XS server 12 mainly includes a XS Request Manager Unit (XSRMU) 30, XS Switch Manager (XSSM) 31, XS Resource Manager XSRM 32, XS Main Manager (XSMM) 33 and resource connection 34. In the XS server 12, the client 11 requests are received by the XSRMU 30, that handles the server communication part.
- 30 Depending on the type of the request, the XSRMU 30 will distribute the requests to the XSSM 31 or the XSRM 32 via an XSMM 33 synchronized main queue. The resource connection 34

connects resources, such as ISDN Resource, DSP Resource, S-connect Resource, Conference Resources etc.

The XSRM 32 functional block provides functionality to manage, synchronise and co-ordinate  
5 HW resources. It provides a generic high level interface to the HW resources. Preferably, it abstracts the handling and configuration of HW boards, which interfaces the communications networks. Asynchronous events from the HW interfaces are passed by to the XSRMU 30 via the XSMM 33 synchronized main queue. It has for example mechanisms for allocating, deallocating and translation of switching information related to HW resources. XSRM 32 also  
10 provides a high level interface to the HW and is a single service access point to the external communications network interfaces. The XSRM 32 handles the physical switching and the coordination of the HW.

The XSSM 31 handles the resources in a logical level. It manages the relations between the  
15 resources and uses the XSRM 32 to access them. It also uses the XSRMU 30 to pass requests to other XS servers in the system, regarding handling of resources outside a local machine. The single node switching is solved only by using the local XSRM.

XSMM 33 is the main functional block in the XS. It contains the main application and it  
20 initiates and synchronises other units in the system, for example by managing a multi thread based architecture. It provides functionality to send and receive messages between the different functional blocks, e.g. by controlling a thread secure synchronised queue, and invokes each functional block's Request in the order the queue items are received.

25 The main functions of XSSM 31, among others, are to:

- make connection and disconnections,
- allocate, release and reset specific resources,
- remove connections as a result of certain events, such as external hangup.

30 Preferably, the XSSM uses resource tables for executing its tasks. Requests from the XSRMU 30 are sent to XSSM through a queue managed by the XSMM. In order to perform a task for

example making a connection, the XSSM 31 and the XSRM work very intimately. The tasks are divided so that the XSSM 31 handles the logical part of the switching, which means the part where information about resources, such as Bus tables, Connection Tables and Conference Tables (mainly but not exclusively arranged in XSSM), must be updated in order to provide correct information about the actual physical connections. In other word, XSSM 31 logically connects two connection points, which represents an access point to a call, a conference, a DSP or a Logical Address, to each other (through a connection table) and sends commands to XSRM 32 in order to accomplish the physical connection.

- 10 The distributed switching uses the MC1 bus for transferring media streams between the XS servers.

The XSRMU 32 function block provides functionality to handle the communication in the distributed environment. The communication paths are:

- 15
- XSCU (Client) to XS server communication;
  - XS server to XSCU (Client) communication; and
  - XS server to XS server communication.

The multi-node switching is provided in a three stage process:

- 20 – the first stage includes allocation of a MC1 bus resource and sending a XS Server Message (XSMSG) to the other server requesting an execution of a service and all information needed by the other server to fulfill its part of the switching. The XSMSG is the XS server to server interface message which contains request and notification data from one XS server to another.
- 25 – The second stage is when another server receives an XSMSG. Upon reception it will connect the input from the MC1-bus to the output on the relevant connection point via the internal telephony bus. When this is completed, it will send an XSMSG containing a service execution message and all information needed by the originating server to fulfill the second part of the switching.
- 30 – The third stage is when the originating server receives the Service Execution Message. It will then connect the input connection point to the output on the MC1

bus via the internal telephony bus. If this would fail, it will send a rollback request to the other server asking it to remove the previous connection.

XS further consists of a set of message queues and queue managers. XSCU is the part that  
5 exposes the XS API to the clients and communicates with the XS Servers. Each CL 35  
communicates with the XS through the XSCU interface using the XS API 24.

Essentially, each instance of XSCU has information substantially about all the resources used  
by all CL and also their location (connection to a certain XS Server(s)). XSCU receives  
10 information from each XS Server when a resource is initiated or deleted. Based on said  
information, the requests can be routed to the XS Server best suited to handle a request. The  
information is also useful when a XS Server is out of order or not reachable. In the following,  
this information is called the XSCU System Context (XSSC).

15 Requests that create new resources (for example Make Call, Allocate Conference or Allocate  
DSP) are arranged in an arbitrary server queue to distribute the workload over the XS Servers.  
If this request contains a "pointer", the request will be transferred to a server pointed out by the  
pointer. The requests that relate to the entire XS (e.g. commands such as Logout, Reset,  
Shutdown) are sent to all servers.

20

The XS Server sends notifications to the XSCU entity that sent the request. If a resource is  
created or deleted, a Resource Info Notification will be sent to all other XSCUs, to update their  
XSSC. Some notifications can also be sent to the CLs.

25 If the request is an operation on two resources residing in different locations (apparatuses), the  
XSCU will send the request to one of them, and fill in the logical address of the other resource  
from its XSSC (if it is represented by a GUID and not a logical address already). The logical  
address identifies a low level connection point within the XS server, which allows one input  
connection and several output connections.

30

Preferably, each XSCU entity includes information about essentially all the resources and to

which XS Server they are associated to. The information includes all calls, connections, conferences, DSPs created by any CL and their identity (GUID), which implicitly points out in which location they reside. Each XS XSCU has interfaces towards its client and towards the XS Servers. Some messages travel transparently through the XSCU and some are XS Server/XSCU  
5 internal messages not reaching CLs.

Client requests for a resource in the form of a XS Message are sent from CL to XSCU. Preferably, the XS Messages do not contain any information about the server the message is supposed to be sent to. This is actually handled by the API. When the client sends a message,  
10 the XSCU will look up any resource reference in the message in XSSC to determine in which server the resources are located. If the request involves resources in different servers, information will be added in the XS Message, helping the receiving server to handle the request. The Messages to XSCU received from XS Server are analysed and the XSSC is updated accordingly.

15 Advantageously, the XSSC contains tables for all resources and methods for adding, deleting and finding resources.

The XSCU provides functions to handle the interface to the XS server system through a  
20 distributed API. This functionality is based on handling of instances of messages, XS Message, which is, as described above, the XSCU API message containing request data from CL to XS or notification data from XS to CL. The XS Message contains substantially all APIs.

This handling includes instantiation, sending and receiving XS Message instances. The XSCU  
25 can be the complete API and the single service access point to the XS. The general concept of the API is an asynchronous telegram based solution. The physical HW configuration of the server is hidden from the client. The Microsoft ® MessageQueue system (MSMQ) is for example used for network communication. Obviously, other systems for network communication can be used.

30

The XSCU API structure is divided into several logical parts, for example:



- Server Connection: handles system related functionality, e.g. the connection between CL and the XS Server system;
- Switching: handles connections between XS connection points within the XS Servers;
- 5 - Resource: handles allocation and deallocation of specified resources within the XS Servers;
- Call Control: manages call control functions within the XS Servers.
- Status: messages sent asynchronously from the XS to the CLs informing the affected CL about status changes.
- 10 - Signalling: message sending and signal retrieving using a DSP; and
- Messaging: for example playing and recording messages using a DSP. Messages can be read, deleted and also be shared between concurrent CLs.

The XS also includes an interface 36 to external networks, e.g. PSTN, which for example uses the ISDN protocol. The ISDN interface is implemented in the ISDN resource module within the XS server. The system may also include interfaces to telephony operators based on an analogue telephone interface. This interface is also called S-Connect, and it is implemented in the S-Connect resource module within the XS server.

- 20 The XS provides basic client/server connection via login/logout, where every client has there own specified ID (CLID).

The client server communication, which is based on the transaction of XS Messages, is divided into requests and notification, where the requests represent the messages from the CL to the XS and the notifications represent the messages from the XS to the CLs.

25

To obtain server connection the CL must make an instance of the XSCU API, which handles the asynchronous communication between the client and the XS server locally or over a network, e.g. the LAN.

30

A login request will result in retrieving notifications for all incoming calls retrieved by the XS,

and it will enable the client to send requests to XS. If the login fails, a negative status notification message will be received.

Due to the fact that the XS is able to handle the event of one XS Server that is out of function,  
5 which will result in loss of resources and possibly loss of unserved requests, the XSCU is able to inform the clients about which resources that were lost.

The XS provides functionality to allocate and deallocate defined XS resources: for example Conference and DSP.

10

To minimize network load, the "pointer" mechanism is implemented, which means that the resource can be allocated in the same XS Server and the same interface board as the XS Connection Point which it will be connected to, if the optional XS Connection Point is set in a resource allocation message.

15

A conference resource can be allocated with a guaranteed (fixed or varying) number of participants. A conference participant, is represented by a XS Connection Point, which can be a resource connected with a simplex or duplex connection.

20 A DSP can only be either a receiver or a transmitter at the same time. A connection to a DSP can be duplex, but you can only use the DSP in one direction at the time.

Additionally, the XS provides functionality to handle the events of one or many system components going down. The basis for this functionality is a completely distributed and decentralised architecture. The system is able to handle the event of one XS Server that goes  
25 down or is not reachable. This situation will result in loss of resources and possibly loss of unserved requests. The loss will be limited to the resources directly connected to the failing node. XS informs the clients about which resources that were lost.

30 In an embodiment, XSSM 30 also handles conference allocation. If the XS server does not have any free conference resources available, it will pass the request further to the next XS server in

the system. The "next" server is selected by the one that has not read the message from its queue. This process will end when any XS server takes care of the request or if there is no XS server left to send to (that has not read the message from its queue). If there are no conference resources available in the entire system, the last XS server will send a "failed" notification to  
5 the requesting CL

Generally, the connection is made through the following process:

- CL makes a XS Connect request to the XSCU, which distributes it to a proper XS server.
- 10 – The XS server makes a local allocation of a resource, and transfer the request including an INPUT address and resource address to the next affected XS server.
- The latter XS server makes the local switching and transfers the request in a notification, including an OUTPUT address, back to the first XS server.
- The first XS server now makes the local switching and sends a notification to the  
15 CL via XSCU and the resource (connection) is allocated.

Back to fig. 1, a Message Queue Handler provides functionality to send and receive messages to/from message queues and to locate, create and destroy message queues. A Message Queue Handler has its own queue where all incoming messages are read. All outgoing messages are  
20 sent to other queues, associated with other Message Queue Handlers. A Message Queue Handler can be used by a client or a server in the XS. The client users are then the XSCUs and server users are XS Request Managers.

The queues 14 are divided into two categories, server queues and client queues, thus providing  
25 a straightforward way to identify if the queues are associated with a server or with a client.

Obviously, the invention is not limited to the above-described network arrangement but any kind of computer network, such as computer networks, (cellular) telecommunication networks etc., can use the benefits of the invention.

30

The invention is not limited to the shown embodiment but can be varied in a number of ways

without departing from the scope of the appended claims and the arrangement and the method can be implemented in various ways depending on application, functional units, needs and requirements etc.

## CLAIMS

1. An arrangement in a communications network including one or more clients (11) and one or more servers (12) providing services and/or resources (13), at least one of which is accessed by the client (11) for obtaining said resource or service (13),  
5 *characterised in,*  
that said arrangement includes a part (XSCU) (24) arranged for controlling communications between the client and the server (12), said part (24) including an updateable database (XSCU Context) for storing information at least about said resource/service and corresponding server  
10 (12), and that said part is client API.

2. An arrangement according to claim 1,  
*characterised in,*  
that said part is implemented as a distributed, asynchronous, telegram-based client API.

15 3. An arrangement according to claim 1,  
*characterised in,*  
that said part is arranged to distribute client resource/service requests to said servers (12) by managing said database.

20 4. An arrangement according to any of preceding claims,  
*characterised in,*  
that said part also includes server configuration, allowing multi-node switching, load-sharing and error handling.

25 5. An arrangement according to claim 4,  
*characterised in,*  
that said error handling is based on the "graceful degradation" paradigm, where an error just affects the functionality close to the failure.

30 6. An arrangement according to any of preceding claims,

*characterised in,*

that said resource/service is a specified and identified with a Globally Unique Identifier.

7. An arrangement according to any of preceding claims,

5 *characterised in,*

that said database contains information about all resources used by all clients and also their location.

8. A communications network, preferably a computer based network for providing distributed

10 and redundant functionality, e.g. for switching, control and coordination of different

communication protocols interfacing external and/or internal networks (21), said

communications network including at least server units (12) and means (24) for client stations (11) interfacing the server units (12),

*characterised in,*

15 that the communications network further comprises networking arrangement (22) connecting said client stations (11) and said server units (12), said client stations (11) being arranged to

generate messages for requesting services and/or resources (13) from said server units, at least

each server unit being arranged to handle a resource/service, and that said interfacing means is a service access-point, among others including server unit configurations and resource/service

20 states.

9. A network according to claim 8,

*characterised in,*

that said interfacing means allows multi-node switching, load-sharing and error handling.

25

10. A network according to claim 8,

*characterised in,*

that said interfacing means is implemented as a distributed, asynchronous, telegram-based client API.

30

11. A network according to any of claims 8-10,

*characterised in,*

that said server unit is arranged to handle interface means as a resource/service.

12. A network according to any of claims 8-11,

5 *characterised in,*

that network architecture is substantially a decentralized distributed system.

13. A network according to claim 12,

*characterised in,*

10 that the architecture is based on a varying number of server units (12) connected to each other via a Local Area Network (LAN) (22), for example via an IP network such as an ethernet LAN.

14. A network according to claim 8,

*characterised in,*

15 that server units (12) are also connected to each other via a network (23) to achieve switching between the server units.

15. A network according to any of claims 8-14,

*characterised in,*

20 that it is configured using one server unit with a varying number of clients (11).

16. A network according to any of claims 8-14,

*characterised in,*

that it is configured as a distributed multi-server system providing a varying number of clients,  
25 to be connected to a varying number of server units.

17. A network according to claim 11,

*characterised in,*

that Hardware (HW) interfaces are handled as resources, which are encapsulated under a  
30 generic resource interface Manager within a server unit (12).

18. A network according to claim 17,

*characterised in,*

that different resource modules are encapsulated and handled as subsystems under the resource interface.

5

19. A network according to claim 18,

*characterised in,*

that said resource modules include at least one of ISDN Resource module, for handling ISDN (Integrated Services Digital Network) protocol, S-Connect Resource module, for handling

10 analogue operator interface (S-connect), DSP Resource module, for handling DSP (Digital Signal Processing) handling for signalling and messaging, and Conference Resource module, for handling conference devices for conference switching.

20. A network according to any of claims 8-19,

15 *characterised in,*

that said server unit includes a Request Manager Unit (XSRMU) (30), Switch Manager unit (XSSM) (31), Resource Manager Unit (XSRM) (32), Main Manager (XSMM) (33) and resource connection (34), wherein XSRM (32) provides functionality to manage, synchronise and co-ordinate HW resources, XSSM (31) handles the resources in a logical level and manages  
20 the relations between the resources and uses the XSRM (32) to access them, XSMM (33) contains main application and initiates and synchronises other units in the system, and XSRMU (32) provides functionality to handle the communication in the distributed environment.

21. A network according to any of claims 8-20,

25 *characterised in,*

that the network is an Exchange System (20).

22. A method for providing a resource and/or service to a client in a communications network including servers managing said resource/service,

30 *characterised in,*

that the method comprises the steps of:



22.

- providing means (XSCU) (24) for controlling communication between the client and the server (12), said means including an updateable table (XSCU Context) for storing information at least about said resource/service and corresponding server (12),
- 5 – controlling said table when sending a request for said resource/service, and
- based on information retrieved from said table, routing said request to a server suited to handle the request.

23. A method according to claim 22,

10 *characterised in,*

that in the server (12), the client requests are received by a Request Manager (30), which handles the server communication part, and depending on the type of the request, the Request Manager (30) distributes the requests to a Switching manager or Resource Manager (32) via a main queue.

15

24. A method according to claim 23,

*characterised in,*

that the switching is multi-node and provided in several stages, including:

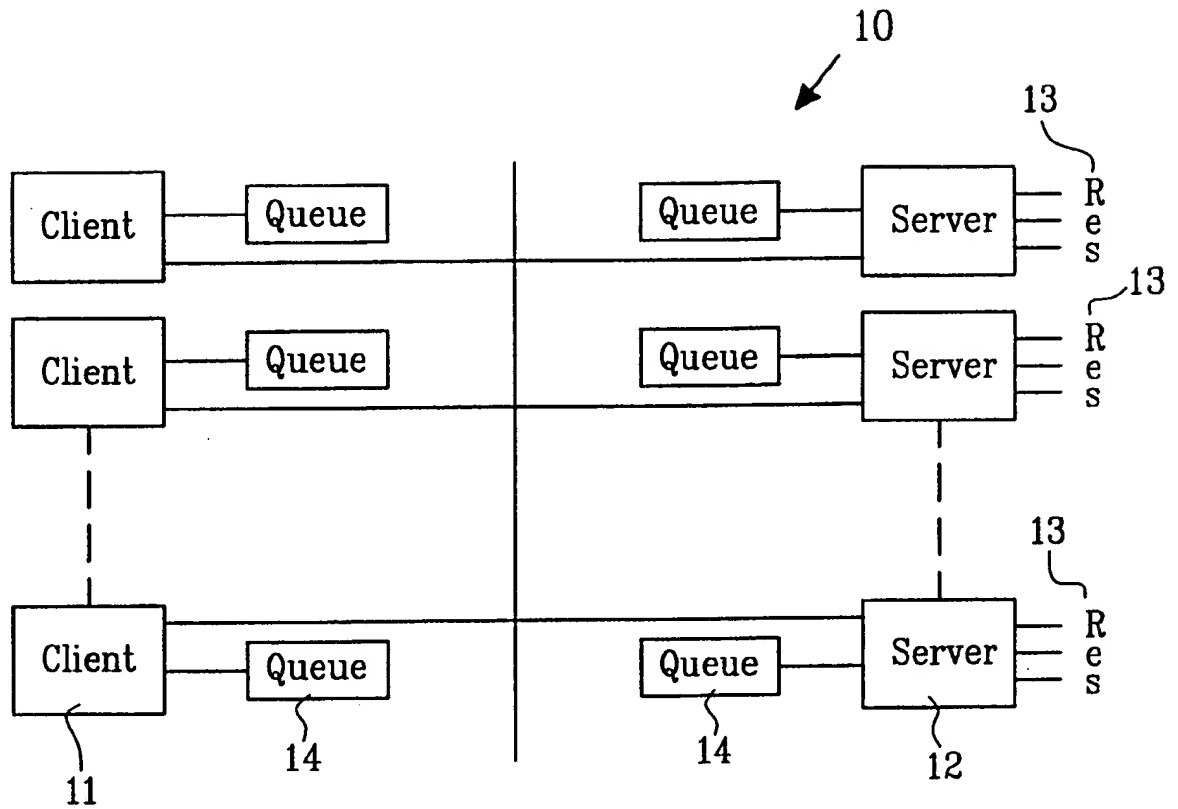
- (a) allocation of a connection means (23) resource and sending a Message
- 20 (XSMSG) to other servers requesting an execution of a service and all information needed by the other server to fulfill its part of the switching,
- (b) when another server receives the Message (XSMSG), connecting the input from the connection means to an output on a relevant connection point
- (c) sending an XSMSG containing a service execution message and all
- 25 information needed by the originating server to fulfill stage (b),
- (d) when the originating server receives a Service Execution Message, connecting the input connection point to an output on the connection means.

25. A method according to claim 24,

30 *characterised in,*

that client request, which creates new resources are arranged in an arbitrary server queue to

distribute the workload over the servers and if this request contains a pointer, transferring the request to a server pointed out by the pointer.



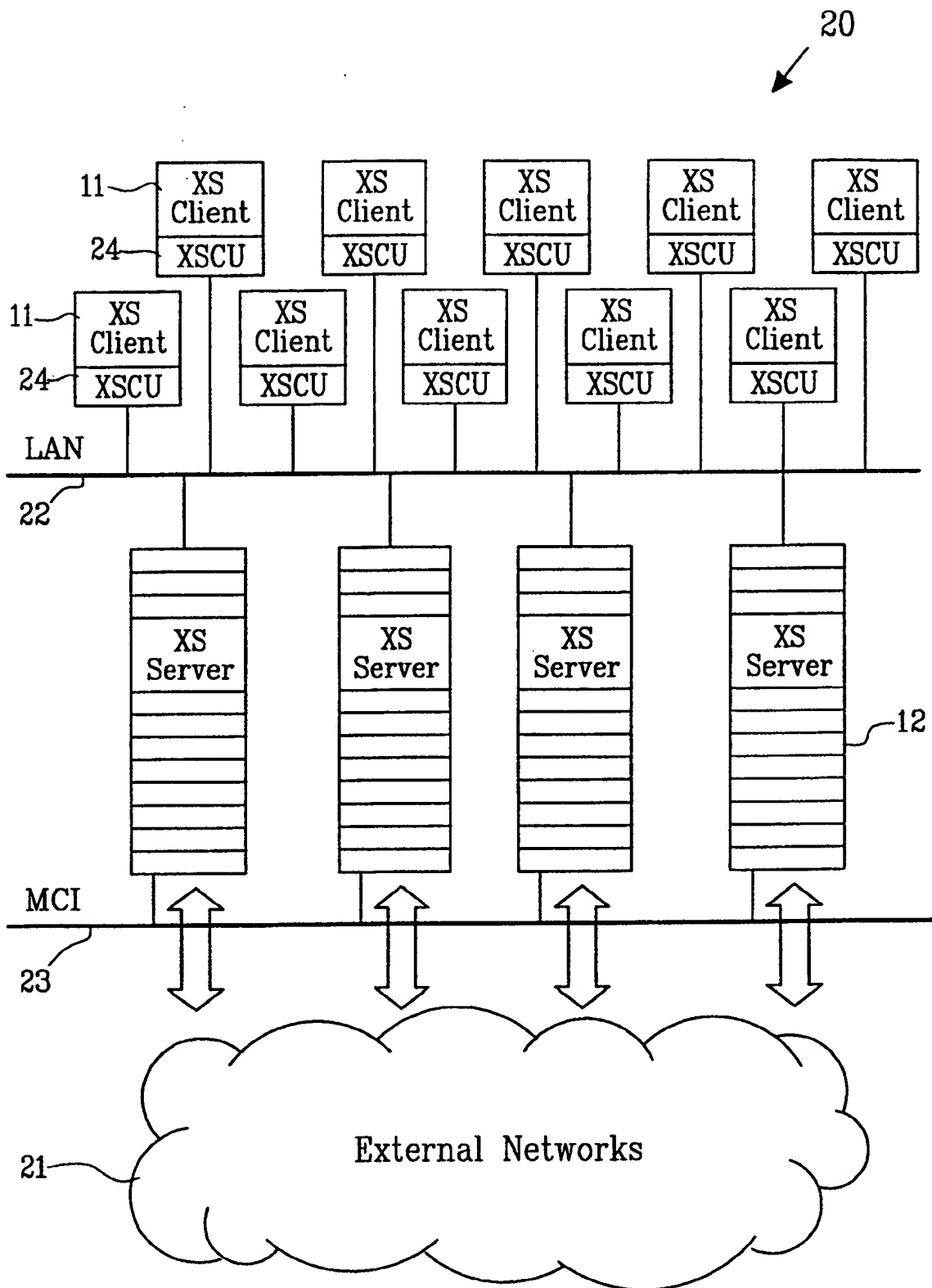
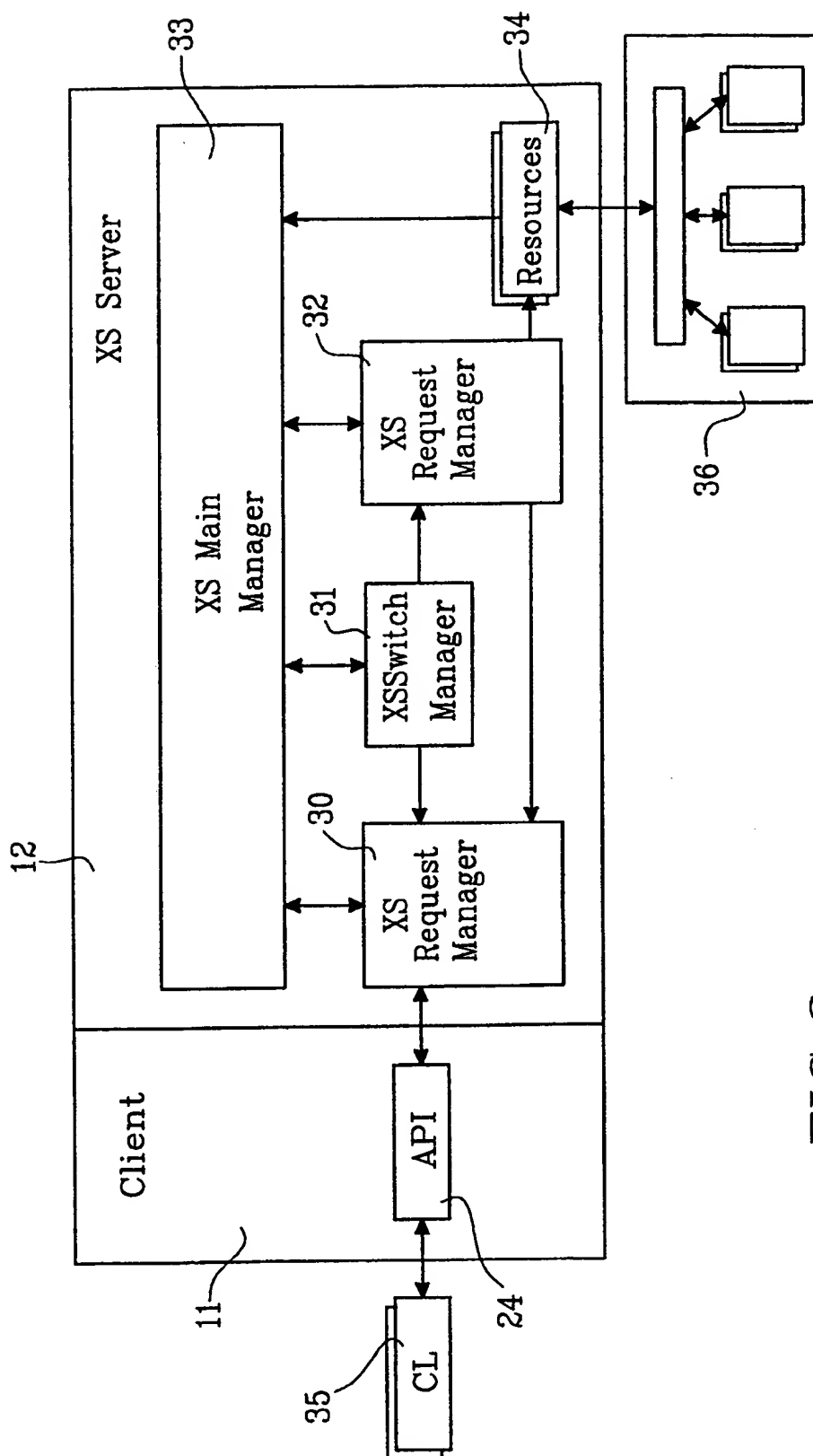


FIG.2



**FIG. 3**

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/SE 99/02247

## A. CLASSIFICATION OF SUBJECT MATTER

IPC7: G06F 17/30, H04L 12/00 // G06F 9/46

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: G06F, H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5548723 A (CHRISTOPHER E. PETTUS), 20 August 1996 (20.08.96), column 2, line 21 - line 49; column 11, line 20 - column 12, line 19; column 15, line 17 - column 16, line 3  --	1-25
Y	US 5101348 A (ANDREW H. ARROWOOD ET AL), 31 March 1992 (31.03.92), column 1, line 50 - column 2, line 8; column 3, line 1 - line 30; column 10, line 45 - line 66, abstract, column 11, line 10 - line 25  --	1-25

☒ Further documents are listed in the continuation of Box C.
 ☒ See patent family annex.

* Special categories of cited documents:	"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"B" earlier document but published on or after the international filing date	"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"R" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search	Date of mailing of the international search report
25 February 2000	10 -04- 2000
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No: +46 8 666 02 86	Authorized officer Björn Edlund/CL Telephone No. +46 8 782 25 00

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/SE 99/02247

## C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5475819 A (STEVEN P. MILLER ET AL), 12 December 1995 (12.12.95), column 1, line 66 - column 2, line 2; column 3, line 5 - column 4, line 1; column 6, line 4 - line 48  --	1,8,22
A	US 5796952 A (OWEN DAVIS ET AL), 18 August 1998 (18.08.98), see whole document  --	1,8,22
A	US 5574904 A (HIDEO YUNOKI ET AL), 12 November 1996 (12.11.96), see whole document  -- -----	1,8,22

## INTERNATIONAL SEARCH REPORT

Information on patent family members

02/12/99

International application No.

PCT/SE 99/02247

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5548723 A	20/08/96	CA 2178580 A	22/06/95
US 5101348 A	31/03/92	DE 68918765 D,T	27/04/95
		EP 0348331 A,B	27/12/89
		JP 2038773 C	28/03/96
		JP 2044944 A	14/02/90
		JP 7075354 B	09/08/95
US 5475819 A	12/12/95	CA 2048306 A	03/04/92
		DE 69122830 D,T	28/05/97
		EP 0479660 A,B	08/04/92
		JP 4230567 A	19/08/92
US 5796952 A	18/08/98	AU 6573198 A	20/10/98
		WO 9843380 A	01/10/98
US 5574904 A	12/11/96	JP 2532148 B	11/09/96
		JP 3214851 A	20/09/91